WHAT IS CLAIMED IS:

1. A method comprising,

in a network, encapsulating data requests generated by an application in a first system;

transferring the encapsulated data requests to a second system;

executing the encapsulated data requests in the second system; and

processing in the first system responses generated by the encapsulated data requests in the second system.

2. The method of claim 1 in which encapsulating comprises:

generating an Extensible Markup Language (XML) structure for each data request; and

converting the XML structure to an XML request.

3. The method of claim 2 in which the XML structure comprises a variable stream of data stored in memory of the first system, the stream including an XML element for each request.

4. The method of claim 3 in which the XML element is a class object whose data is stored to generate XML.

5. The method of claim 4 in which the XML element includes data from a data set object.

6. The method of claim 5 in which the data set object includes table dictionaries, column names and data from record sets, and stored procedure parameters.

7. The method of claim 1 in which transferring includes a text transmission protocol.

8. The method of claim 7 in which the text transmission protocol is Hypertext Transfer Protocol.

9. The method of claim 1 in which executing comprises:

de-encapsulating the encapsulated data requests by parsing into request statements; and

executing the request statements.

10. The method of claim 9 further comprising:

translating responses from the executed request statements into an XML format; and

sending the XML formatted responses to the first system.

11. A distributed application method comprising:

converting application requests in a first system;

transmitting the converted application requests to a second system over a network;

parsing the converted application requests in the second system into request statements; and

executing the request statements in the second system.

12. The method of claim 11 in which converting comprises:

generating a data structure for storing data and parameters related to an application that produced the application requests;

translating the application requests into a standardized delimited data structure stored in a memory of the first system; and

transforming the standardized delimited data structure in conjunction with the data structure into a stream of text based data utilizing a Extensible Markup Language (XML) format.

13. The method of claim 11 in which the parsing comprises:

breaking down the converted application requests to an executable command format utilizing data and parameters related to an application.

14. The method of claim 13 in which executing further comprises evaluating executable commands prior to execution in the second system.

15. The method of claim 14 in which executing further comprises evaluating results generated by the executable commands.

16. The method of claim 15 further comprising:

converting the results into a stream of text based data in a standardized XML format; and

transmitting the converted results over the network to the first system.

17. An application server method comprising:

generating a first data structure for storing data and parameters related to an application residing in the server;

translating application requests from the application into a delimited second data structure stored in a memory;

generating a stream of text-based data in an Extensible Markup Language (XML) format from the second data structure.

18. The method of claim 17 in which the first data structure includes database tables, procedure results from logic calls and status/error messages.

19. The method of claim 17 in which the second data structure includes an element for each of the application requests.

20. The method of claim 19 in which the element is a class object.

21. A method comprising:

in a server, receiving a stream of text-based data in an Extensible Markup Language (XML) format;

parsing the stream into request statements; and

executing each of the request statements.

22. The method of claim 21 in which executing further comprises intercepting the request statements prior to execution and applying additional logic based on a type or content of the request statements.

23. The method of claim 21 in which executing further comprises applying additional logic to responses generated from executing the request statements.

24. The method of claim 21 further comprising:

converting responses generated from each of the executed request statements into an XML format.

25. A computer program product residing on a computer readable medium having instructions stored thereon which, when executed by the processor, cause the processor to:

convert application requests in a first system;

transmit the converted application requests to a second system over a network;

parse the converted application requests in the second system into request statements; and

execute the request statements in the second system.

26. A computer program product residing on a computer readable medium having instructions stored thereon which, when executed by the processor, cause the processor to:

generate a first data structure for storing data and parameters related to an application residing in the server;

translate application requests from the application into a delimited second data structure stored in a memory;

generate a stream of text-based data in an Extensible Markup Language (XML) format from the second data structure.

27. A computer program product residing on a computer readable medium having instructions stored thereon which, when executed by the processor, cause the processor to:

receive a stream of text-based data in an Extensible Markup Language (XML) format;

parse the stream into request statements; and

execute each of the request statements.

28. An enhanced graphical user interface (GUI) method comprising:

displaying a plurality of visual controls on an input/output device; and

displaying at least one data enabled control on the input/output device.

29. The method of claim 28 in which the data enabled control comprises a control having properties describing data relationships to the control.

30. The interface of claim 29 in which the data enabled control further comprises properties describing locations of data and data sources pertaining to the control.

31. The method of claim 28 in which the data enabled control is user-configurable.

32. The method of claim 30 in which the properties comprise:

a location of a database table;

a name of the database table; and

a column name representing the control.

33. The method of claim 32 in which the properties further comprise:

a listing of table relationships;

an indicator to indicate whether the control is a key column in the table; and

an indictor to indicate whether the control is a primary key column.

34. The method of claim 33 in which the properties further comprise:

an indicator to indicate whether the control is part of a compound primary key;

an indicator to indicate whether a record is locked when in use; and

an indicator to indicate whether the control if data in the control has changed.